REMARKS

In the Office Action, the Examiner indicated that claims 1 through 25 are pending in the application and the Examiner rejected all claims.

The §101 Rejection

On page 2 of the Office Action, the Examiner has rejected claims 1-9, 12, 14-17, and 20-25 under 35 U.S.C. §101 as being directed to non-statutory subject matter. Specifically, the Examiner asserts that these claims are directed to abstract ideas that would not result in a practical application producing a tangible result. Applicant respectfully disagrees. Each of the claims in question relate to the prohibition of loading and storing of new application components on a portable device, thereby managing the use of run-time resources. This tangible result is achieved through the performance of one or more steps to determine and identify certain parameters, the value of which are used to determine if the new programs should or should not be stored on the device. By doing so, the device is assured of running efficiently and with minimal disruption. The Examiner is respectfully requested to reconsider and withdraw the rejection of claims 1-9, 12, 14-17, and 20-25 under 35 U.S.C. §101.

Claim Rejections, 35 U.S.C. §103

On page 3 of the Office Action, the Examiner rejected claims 1, 4, 14 and 20 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 6,910,210 to Chew in view of U.S. Patent No. 6,029,000 to Woolsey et al. and further in view of U.S. Patent No. 5,826,082 to

Bishop et al. On page 5 of the Office Action, the Examiner rejected claims 2-3, 15-16, and 21-22 under 35 U.S.C. §103(a) as being unpatentable over Chew in view of Woolsey et al. and further in view of Bishop et al. and further in view of U.S. Patent No. 6,282,561 to Jones et al. On page 7 of the Office Action, the Examiner rejected claims 5-13, 17-19, and 23-25 under 35 U.S.C. §103(a) as being unpatentable over Jones et al. in view of Chew.

The Present Invention

The present invention provides, for use in a portable device, a resource management method, system, and product that insures that sufficient runtime resources (i.e., volatile memory such as RAM) are available for running a new application component scheduled to be installed and stored on the portable device, before it is stored on the device. When an attempt is made to store the new application component on the portable device (e.g., in non-volatile memory such as flash memory), the storage attempt is blocked unless sufficient runtime resources are available for use by that application component and for all of the application components already stored on the portable device to be run simultaneously.

When determining the amount of runtime resources available to be used by the to-be-stored application, the invention assumes that all existing programs already stored on the portable device are using the maximum amount of runtime resources that they need. By storing a new application component on the portable device only if sufficient runtime system resources are currently available so that all stored applications could be properly run simultaneously; reserving runtime system resources when the new application component is stored; and running stored

application components using only the amount of runtime system resources reserved for those stored application components, the present invention insures that each stored application component will always have a sufficient amount of runtime resources to execute properly. Accordingly, the present invention prevents even the possibility of improper operation of stored application components due to running too many application components simultaneously, poorly designed application components, and/or destructive application components.

U.S. Patent No. 6,910,210 to Chew

U.S. Patent No. 6,910,210 to Chew ("Chew") teaches the concept of preventing the launch of a program if there are insufficient resources available to run the program. Chew determines the availability of resources by polling for available memory or the number of processes currently running before allowing the program to execute (column 9, lines 8-19).

U.S. Patent No. 6,029,000 to Woolsey et al.

U.S. Patent No. 6,029,000 to Woolsey et al. ("Woolsey") teaches a mobile communication system with a cross compiler and cross linker. The Examiner relies upon Woolsey for its teachings at column 21, lines 9-14. Specifically, the portion relied upon by the Examiner states as follows:

> "In step 118, the attributes are retrieved from the Bean 90 and in step 120 the applet determines whether the appropriate processor has sufficient resources to run the code. If not, the exception processing step 114 may decline to install the native code, or steps may be taken to free resources."

U.S. Patent No. 5,826,082 to Bishop et al.

U.S. Patent No. 5,826,082 to Bishop et al. ("Bishop") teaches a method for managing resources, in a computer system, allocated to an operation of a running program (see Abstract, and column 3, lines 1-4). The Examiner relies on Bishop for an alleged teaching of "determining the currently available runtime resource by assuming already loaded application components are using the maximum amount of runtime resources reserved for their use", citing column 3, line 55 through column 4, line 24.

U.S. Patent No. 6,282,561 to Jones et al.

U.S. Patent No. 6,282,561 to Jones et al. ("Jones") teaches a resource management mechanism to ensure that real-time application programs running on a single machine or set of machines exhibit predictable behavior. Jones et al. is specifically concerned with managing the loading of already-stored programs into RAM when there are not enough run-time resources available, i.e., Jones may block the run-time operation of a program already stored on a device if there are insufficient run-time resources to properly run that program. Various options exist in Jones, including the ability to run a program at a diminished capacity using less than the optimal quantity of run-time resources and also includes a dynamic feedback mechanism for initiating renegotiation of run-time resource reservations when appropriate.

The Examiner has not Established a *prima facie* Case of Obviousness

Applicant notes that the rejections under 35 U.S.C. §103(a) presented in the present Office Action are, with one exception, identical to the rejections presented in the previous Office Action. The exceptions relate to the addition of the Bishop patent, and the details of the new grounds for rejection are set forth in paragraphs 8 and 9 of the present Office Action. Accordingly, while the Applicant reiterates the previously-presented arguments, the focus of the present response is on the newly-cited Bishop patent.

As set forth in the MPEP:

> To establish a *prima facie* case of obviousness, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skilled in the art, to modify the reference or to combine reference teachings.
> MPEP 2143

As noted above, the present invention blocks completely the storage on a portable device of a program that a user is attempting to store on the portable device, if the possibility exists that there will not be sufficient run-time resources available on the portable device to run the program properly. More specifically, a determination is made as to the amount of run-time resources required to run all programs currently stored on the portable device at their maximum level (referred to as the CARSRMAX), and a determination of the quantity of run-time resources that would remain available if such already-stored programs were operating in this manner. The present invention also determines the maximum amount of run-time resources required of the to-be-stored application, and if there are sufficient available run-time resources for all of the programs stored on the portable device and the to-be-stored program to run at their full capacity

simultaneously, only then is the to-be-stored application allowed to be stored on the portable device. If there are insufficient run-time resources, either the to-be-stored application is blocked from being stored on the portable device, or one or more of the currently-stored applications is removed to assure that there will be sufficient run-time resources for the remaining stored applications and the new to-be-stored application to run properly.

These aspects of the present invention are claimed in two basic formats, represented by independent claims 1 and 5. In claim 1, the following language is expressly claimed:

> "determining a CARSRMAX (Currently Available Runtime System Resources of the portable device assuming already loaded application components are using the MAXimum amount of runtime resources reserved for their use) of said portable device;
>
> comparing, using said processor, said maximum required runtime resources for said one or more new application components to said CARSRMAX; and
>
> prohibiting, using said processor said one or more new application components from being loaded and stored on said portable device if said CARSRMAX is less than said maximum required runtime resources."

Claim 5 states:

> "A runtime resource management method for use with a portable device, said method comprising the step of:
>
> reserving maximum runtime resources required for each application component stored in flash memory of the portable device."

With respect to claim 1 (as well as independent claims 14 and 20), the Examiner relies on a combination of Chew, Woolsey, and Bishop in rejecting the claims. As noted above, Chew teaches the well known concept of precluding the launching of a program if there is not sufficient runtime memory available to run the program that a user is attempting to launch. Nowhere in

Chew is there a teaching of determining the currently available runtime system resources of the portable device assuming already loaded application components are using the maximum amount of runtime resources reserved for their use (the CARSRMAX), and despite assertions that it does, the Examiner has not shown where in Chew such a teaching exists. Rather, Chew simply looks at how much of the run-time memory is currently being used, and then if there is sufficient additional run-time memory available to run the program being launched, it will be allowed to be launched; otherwise, it will be prevented from being launched. Thus, using the teachings of Chew, it is possible and likely that there may be significantly more programs loaded on the computer than can actually be run concurrently.

The Examiner relies on Woolsey to provide a teaching that one or more new application components will be prevented from being loaded and stored in the flash memory if the CARSRMAX is less than the maximum required runtime resources available. Contrary to the assertions of the Examiner, Woolsey, like Chew, says nothing about determining the CARSRMAX. Rather, Woolsey is only concerned with whether the processor has sufficient resources to run the code. Woolsey contains no teaching or suggestion of determining if a program being installed would have the necessary runtime resources available if all installed programs were also using their maximum runtime resources at the same time.

The addition of Bishop, similarly, provides no such teaching or suggestion. Bishop teaches that by reserving a part of a runtime resource for a request prior to allocating the resource to the request, the thread requesting the resource is certain to have enough of the resource to complete its operation (see column 3, line 31-36 of Bishop). Although the present invention is

secondarily concerned with the issue of resource allocation, it is primarily concerned with preventing a portable device from ever storing programs that, if all running simultaneously, would exceed the runtime resources available on the portable device. In other words, there is really no need to worry about allocation at all, since use of the present invention prevents there from ever being a situation where runtime resources are unavailable.

Accordingly, since these elements are specifically claimed in independent claims 1, 14, and 20, and since the prior art contains no teaching or suggestion of these aspects, it is respectfully submitted that the claims patentable define over the combination of Chew, Woolsey, and Bishop proposed by the Examiner.

With respect to independent claims 5, 17, and 23, as set forth above, these claims each require that the maximum runtime resources required for each application component stored in flash memory of a portable device be reserved. Jones contains no such teaching. Jones may block the runtime operation of a program already stored on a device if there are insufficient runtime resources to properly run that program. However, the resource manager of Jones cannot block the storage of new applications in the non-volatile memory (e.g., the hard drive or flash memory) of a system utilizing the Jones resource manager, and this concept is not taught or suggested by Jones. Jones simply manages the use of run-time resources by any programs stored on the fixed storage memory; the number, size, etc. of programs in the fixed storage memory of Jones is irrelevant to the disclosure of Jones.

Accordingly, applicant asserts that the claims, as amended, patentably define over Chew and/or Chew combined with Woolsey, Bishop and/or Jones. Thus, the claims are in condition for

allowance. Reconsideration of the claims and an early Notice of Allowance are earnestly solicited.

Conclusion

The present invention is not taught or suggested by the prior art. Accordingly, the Examiner is respectfully requested to reconsider and withdraw the rejection of the claims.

The Commissioner is hereby authorized to charge any fees associated with this communication to Deposit Account No. 09-0461.

Respectfully submitted

November 1, 2006
Date

/Mark D. Simpson/
Mark D. Simpson, Esquire
Registration No. 32,942

SYNNESTVEDT & LECHNER LLP
2600 ARAMARK Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189